

FIG. 2

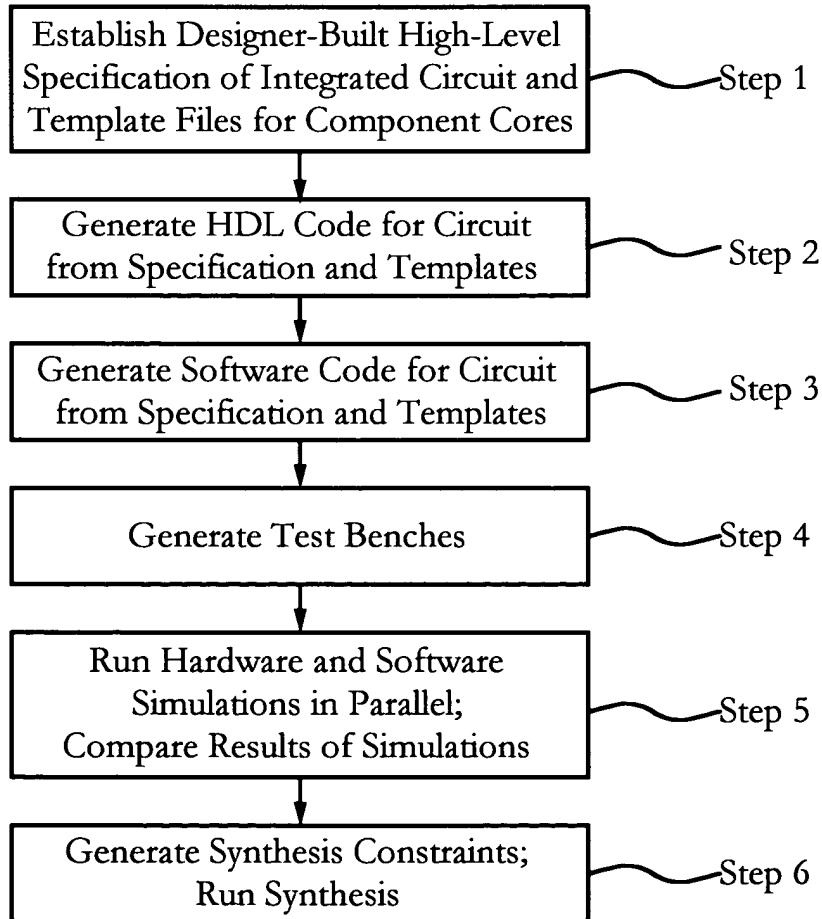


FIG. 3-B

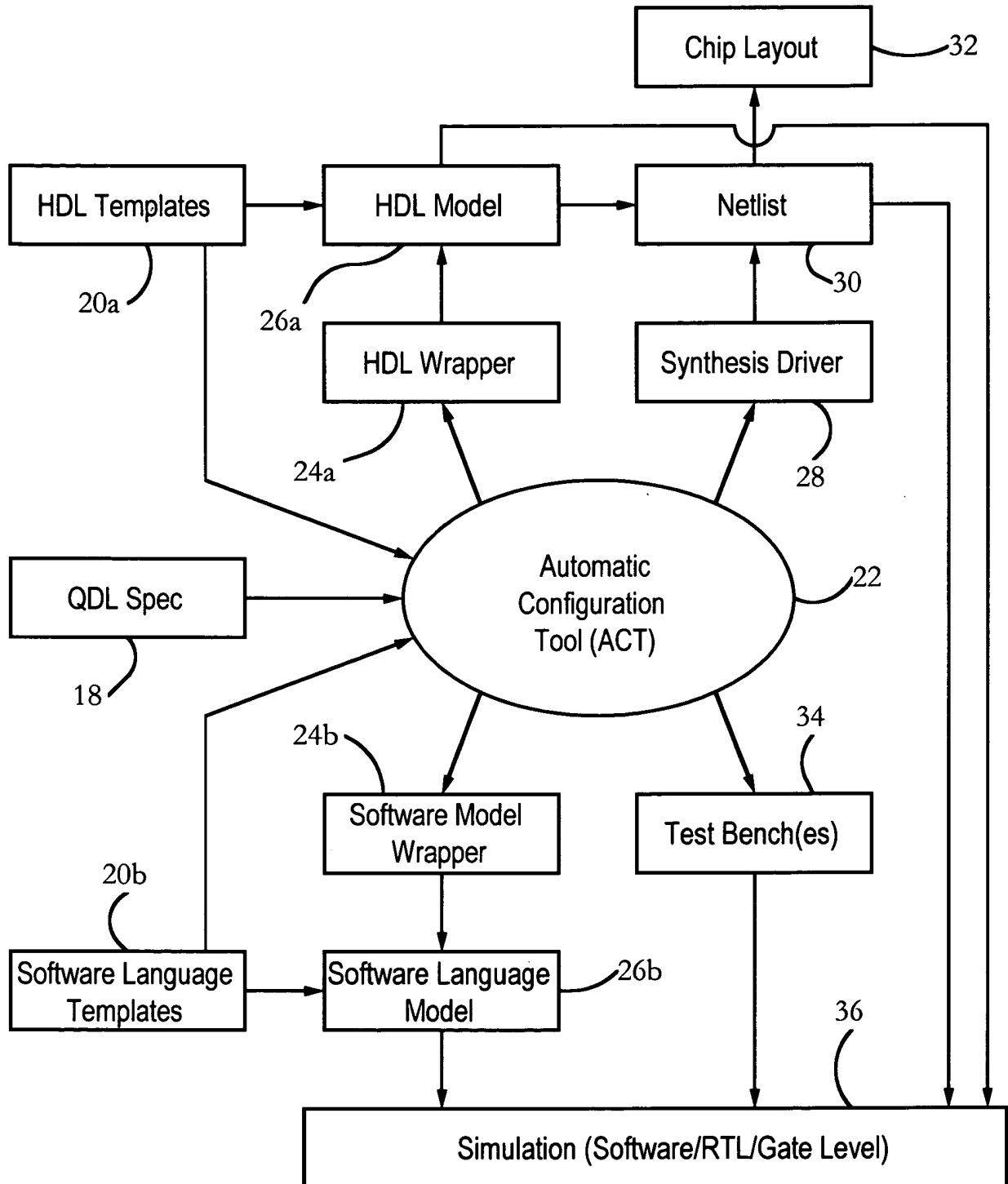


FIG. 3-A

40

FIG. 5

50

60a

FIG. 6-B

```

(* Start: QDL_PORT_LIST */
clk
rst_
vp_bs_id
vp_bs_data
vp_bs_rdy
vp_bs_req
vp_mvp_mpeg
vp_mvp_layer
vp_mvp_ext
vp_mvp_code
vp_mvp_data
vp_mvp_rdy
vp_mvp_req
/* End: QDL_PORT_LIST */);

```

```
/* Start: QDL_PARAM_LIST */
```

FIG. 6-C

```
parameter BSN = 'MVP_BSN';
parameter BSW =
((BSN)<=2)?1:(((BSN)<=4)?2:(((BSN)<=8)?3:(BSN)<=16)?4:5));
parameter MVP_NR = 1
/* End: QDL_PARAM_LIST */
```

60c

```
/* Start: QDL_BUS_DEFS */
```

FIG. 6-D

```
parameter BS_ID_MSB = (0);
parameter BS_ID_LSB = (0);
parameter BS_ID_W = (BS_ID_MSB - BS_ID_LSB + 1);
parameter BS_DATA_MSB = (16-1);
parameter BS_DATA_LSB = (0);
parameter BS_DATA_W = (BS_DATA_MSB - BS_DATA_LSB + 1);
parameter BS_ALL = BS_ID_W + BS_DATA_W;
parameter MVP_MPEG_MSB = (0);
parameter MVP_MPEG_LSB = (0);
parameter MVP_MPEG_W = (MVP_MPEG_MSB - MVP_MPEG_LSB + 1);
//some parameters cut out
parameter MVP_DATA_MSB = (15);
parameter MVP_DATA_LSB = (0);
parameter MVP_DATA_W = (MVP_DATA_MSB - MVP_DATA_LSB + 1);
parameter MVP_ALL = MVP_MPEG_W + MVP_LAYER_W +
                    MVP_EXT_W + MVP_CODE_W + MVP_DATA_W;
/* End: QDL_BUS_DEFS */
```

60d

60g

```
q1_qi #(BS_ALL, 1) qa_mv16_qi (
```

FIG. 6-G

```
.clk (clk ),
.rst_ (rst_ ),
.idata ({vp_bs_id, vp_bs_data}),
.irdy (vp_bs_rdy),
.ireq (vp_bs_req),
.odata ({icmd, idata}),
.ordy (irdy),
.oreq (ireq,.);
```


FIG. 7-A

-70a

FIG. 7-B

-70b

```

namespace QuArc {
bool    QDL_NAME::sim_logic(void)
{
    //bit-accurate model of algorithm here
    return true;}
QDL_NAME::QDL_NAME (const QDL_NAME\_FDS&  arg_fds)
:
    Qblock          ( );
    p_f_des          (arg_fds),
    /* QDL_CONST_INIT */
{ //initialization code here }
bool QDL_NAME::sim_core(void)
{
    /* QDL_INPUT_CONNECTIONS */
    /*QDL_OUTPUT_CONNECTIONS*/
    return sim_logic();} //End Namespace

```

FIG. 7-C

70c

```

namespace QuArc {
bool QAMVP16::sim_logic(void)
{
    //bit-accurate model of algorithm here
    return true;}
QAMVP16::QAMVP16(const QAMVP16_FDS& arg_fds)
:
    Qblock          ( );
    p_f_des          (arg_fds),
    /* Start: QDL_CONST_INIT */
    p_vp_bs          ( ),
    p_vp_mvp          ( )
    /* End: QDL_CONST_INIT */
{ //initialization code here }
bool QAMVP16::sim_core(void)
{
    /* Start: QDL_INPUT_CONNECTIONS */
    p_vp_bs.qpipe (& p_f_des.vp_bs_fds);
    p_vp_bs.instance_name (p_f_des.vp_bs_fds.instance_name());
    /* End: QDL_INPUT_CONNECTIONS */

    /* Start: QDL_OUTPUT_CONNECTIONS */
    p_vp_mvp.qpipe (& p_f_des.vp_mvp_fds);
    p_vp_mvp.instance_name(p_f_des.vp_mvp_fds.instance_name());
    /* End: QDL_OUTPUT_CONNECTIONS */
    return sim_logic(); }

```

FIG. 7-D

70d

-80a

80b

```

instantiate      {object      = qa_miqc;
                  name        = iqc;
                  REQUIRED_CONN;
                  connect {    type      = iqz;}
                  connect {    type      = dat;
                                var DW    = 16;}}
RAM_1r1w_connect(z, z, 6, 12);}}

```

90a

FIG. 9-A

```

qa_miqa # (    BSN,
               BSW,
               1 )

iqa ( .clk      ( clk ),
      .rst_     ( rst_ ),
      .mvp_mpeg ( vp_mvp_mpeg),
      .mvp_layer ( vp_mvp_layer ),
      .mvp_ext  ( vp_mvp_ext ),
      .mvp_code ( vp_mvp_code ),
      .mvp_data ( vp_mvp_data ),
      .mvp_rdy  ( vp_mvp_rdy [0:0] ),
      .mvp_req  ( vp_mvp_req [0:0] ),
      .iqz_as   ( iqz_as ),
      .iqz_data ( iqz_data ),
      .iqz_rdy  ( iqz_rdy ),
      .iqz_req  ( iqz_req ),
      .qaddr    ( qaddr ),
      .qrde     ( qrde ),
      .qrdata   ( qrdata ),
      .qwre     ( qwre ),
      .qwdata   ( qwdata ) );

qa_miqc # (    DAT_NR )
iqc ( .clk      ( clk ),
      .rst_     ( rst_ ),
      .iqz_as   ( iqz_as ),
      .iqz_data ( iqz_data ),
      .iqz_rdy  ( iqz_rdy [0:0] ),
      .iqz_req  ( iqz_req [0:0] ),
      .dat_data ( dat_data ),
      .dat_rdy  ( dat_rdy ),
      .dat_req  ( dat_req ),
      .zrde     ( zrde ),
      .zraddr   ( zraddr ),
      .zrdata   ( zrdata ),
      .zwre     ( zwre ),
      .zwaddr   ( zwaddr ),
      .zwdata   ( zwdata ) );

//RAM Instantiations Here;

```

90b

FIG. 9-B

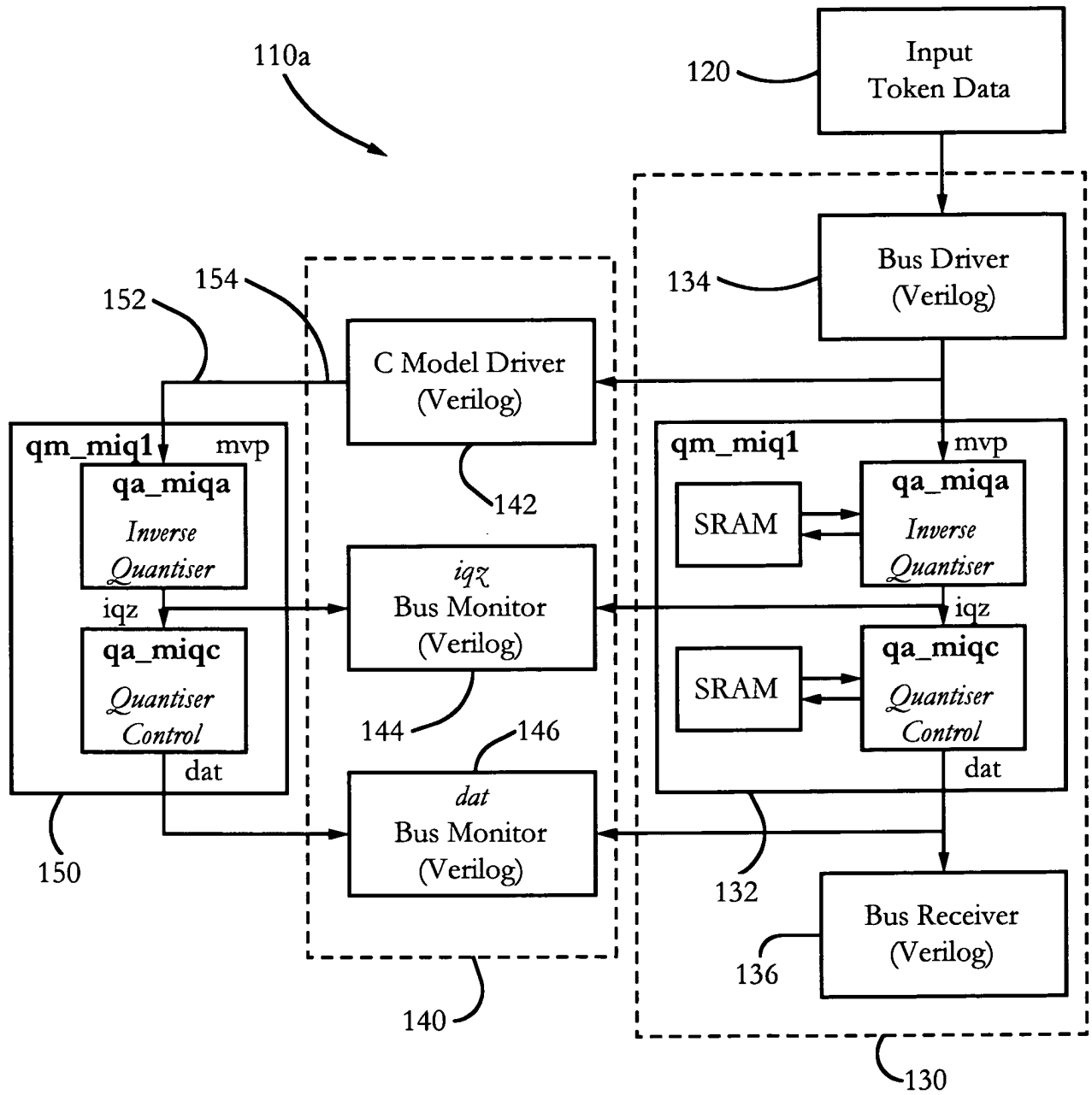


FIG. 11-A

320a

320b

```

proc QsynSetTokenInConstraint { bus } {
    upvar token_input_delay token_output_delay ...(rest of variables)
    set token_ports [get_ports "${bus}*"]
    set in_ports [filter $token_ports "@port_direction == in"]
    set out_ports [filter $token_ports "@port_direction == out"]
    set_input_delay $token_input_delay -clock $clock $in_ports
    set_output_delay $token_output_delay -clock $clock $out_ports
    set_input_delay $if_input_delay -clock $clock $(bus)rdy
    set_output_delay $if_output_delay -clock $clock $(bus)req
    set_load $qif_load $out_ports
    set_drive $def_drive $in_ports
    group_path -critical_range $token_crit_range -name $bus -to $out_ports
    unset token_ports
    unset in_ports
    unset out_ports
    return 1}

```

FIG. 12-B